

IN THE CLAIMS

Please cancel Claims 1-25 and add new Claims 26-49 as follows:

---

1-25. Cancelled.

26. (New) A method for detecting corruption associated with a stack in a storage device,  
the method comprising the steps of:

storing a first predetermined value in a first address location immediately preceding the  
starting location of the stack;

detecting the occurrence of a stack operation within the stack; and

comparing the value in the first address location to the first predetermined value to determine  
if the stack operation corrupted the first predetermined value stored in the first address location.


27. (New) The method as set forth in Claim 26, wherein the first predetermined value  
comprises a known bit pattern.

28. (New) The method as set forth in Claim 26, wherein the first predetermined value  
comprises a processor readable address.

29. (New) The method as set forth in Claim 26, wherein the first predetermined value comprises a processor readable instruction.

30. (New) The method as set forth in Claim 26, wherein the stack operation inserts data in the stack.

31. (New) The method as set forth in Claim 26, wherein the stack operation removes data from the stack.

 32. (New) The method as set forth in Claim 26, further comprising the step of storing a second predetermined value in a second address location immediately following the ending location of the stack.

33. (New) The method as set forth in Claim 32, wherein the second predetermined value comprises a known bit pattern.

34. (New) The method as set forth in Claim 32, wherein the second predetermined value comprises a processor readable address.

35. (New) The method as set forth in Claim 32, wherein the second predetermined value comprises a processor readable instruction.

36. (New) The method as set forth in Claim 32, wherein the stack operation inserts data in the stack.

37. (New) The method as set forth in Claim 32, wherein the stack operation removes data from the stack.

38. (New) A system for detecting corruption associated with a stack, the system comprising:

a processor; and

a storage medium comprising a stack, the stack beginning at a starting location and ending at an ending location, wherein the processor is operable to store a first predetermined value in a first address location immediately preceding the starting location of the stack, detect the occurrence of a stack operation within the stack; and compare the value in the first address location to the first predetermined value to determine if the stack operation corrupted the first predetermined value stored in the first address location.

39. (New) The system as set forth in Claim 38, wherein the first predetermined value

comprises a known bit pattern.

40. (New) The system as set forth in Claim 38, wherein the first predetermined value comprises a processor readable address.

41. (New) The system as set forth in Claim 38, wherein the first predetermined value comprises a processor readable instruction.

42. (New) The system as set forth in Claim 38, wherein the stack operation inserts data in the stack.

43. (New) The system as set forth in Claim 38, wherein the stack operation removes data from the stack.

44. (New) The system as set forth in Claim 38, wherein the processor is further operable to store a second predetermined value in a second address location immediately following the ending location of the stack.

45. (New) The system as set forth in Claim 44, wherein the second predetermined value comprises a known bit pattern.

46. (New) The system as set forth in Claim 44, wherein the second predetermined value comprises a processor readable address.

47. (New) The system as set forth in Claim 44, wherein the second predetermined value comprises a processor readable instruction.

48. (New) The system as set forth in Claim 44, wherein the stack operation inserts data in the stack.

49. (New) The system as set forth in Claim 44, wherein the stack operation removes data from the stack.